

**Namal University, Mianwali**

Department of Electrical Engineering

Communication Systems (Lab)

Lab – 5

**Energy Spectral Density and Power Spectral Density using MATLAB**

Student Name	Student ID
Suleman Ghani	NIM-BSEE-2021-42

Instructor: Dr. Sajjad Ur Rehman

Lab Engineer: Faizan Ahmad

## Introduction

The purpose of this lab is to enable the students to calculate and plot Energy Spectral Density and Power Spectral Density of signals.

## Course Learning Outcomes

CLO2: Develop software simulations to observe the performance of analog and digital communications systems

CLO4: Report desired results, proofs and calculations

## Equipment

- Software
  - Matlab

## Instructions

- This is a group lab. You will perform the tasks in groups of 2 and submit a individual report.
- Please see the rubric at the end of the lab for marks distribution.
- Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, all students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken

## Energy Spectral Density and Power Spectral Density

The energy  $E_g$  of a signal  $g(t)$  is defined as the area under  $|g(t)|^2$ . We can also determine the signal energy from its Fourier transform  $G(f)$  through Parseval's theorem.

The energy of a signal  $g(t)$  is the result of energies contributed by all the spectral components of the signal  $g(t)$ .

$$E_g = \int_{-\infty}^{+\infty} |g(t)|^2 dt$$

$$E_g = \int_{-\infty}^{+\infty} |G(f)|^2 df$$

$|G(f)|^2$  is the energy spectral density (per unit bandwidth in hertz) of  $g(t)$ .

$$ESD = |G(f)|^2$$

So

$$E_g = \int_{-\infty}^{+\infty} ESD df$$

If the signal  $g(t)$  is a power signal, then its power is finite, and the truncated signal  $g_T(t)$  is an energy signal as long as  $T$  is finite.

Then the power of the signal can be defined as:

$$P_g = \int_{-\infty}^{+\infty} \lim_{T \rightarrow \infty} \frac{|G_T(f)|^2}{T} df$$

We define the power spectral density as:

$$PSD = S_g(f) = \lim_{T \rightarrow \infty} \frac{|G_T(f)|^2}{T}$$

$$P_g = \int_{-\infty}^{+\infty} S_g(f) df$$

ESD and PSD both are positive and real and even functions of  $f$ .

## ***Task 1 – Energy and Energy Spectral Density***

The following steps can be used to find the energy spectral density of a signal 'x' using MATLAB:

1. Define  $N = \text{length}(x)$
2. Use  $\text{fft}(x)/fs$  to compute normalized fft of the signal.
3. The above function gives the Fourier transform of the signal, which is not centered at the zero frequency. In order to move the zero frequency component to the center, use  $\text{fftshift}()$ .
4. The Fourier transform is in complex form. Its magnitude spectrum can be obtained by using the  $\text{abs}()$  function.
5. The energy spectral density is simply the squared amplitude spectrum.
6. The total energy of the signal can be obtained by taking the sum of the energy spectral density and multiplying the result with  $fs/N$ .

### ***Exercise 1***

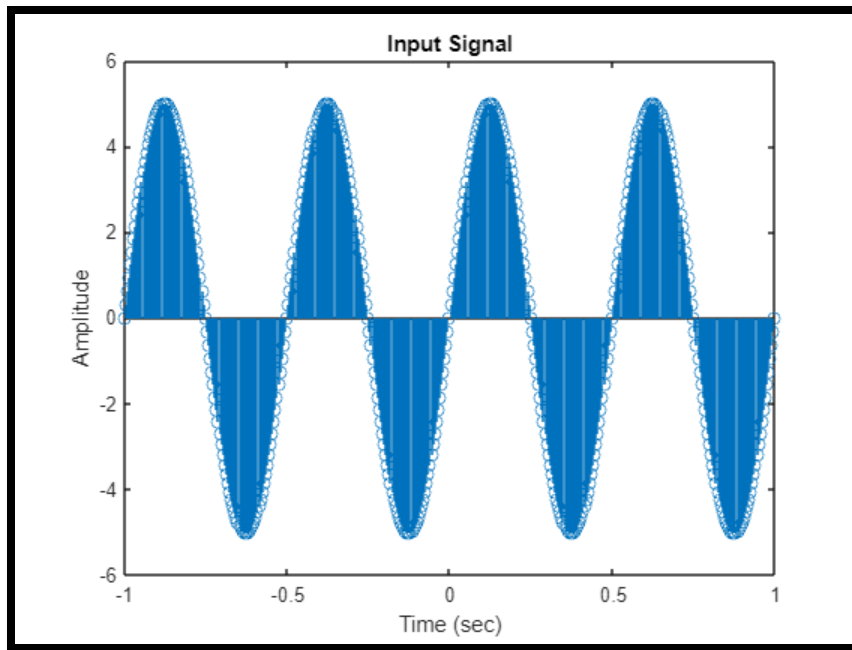
1. Using sampling frequency to be 200, and time increment as  $1/fs$ , generate a sinusoidal signal with amplitude 5 and frequency 2 Hz, between -1 and 1 sec.
2. Add uniform noise using  $\text{randn}$  function. The resulting signal will be used as input signal. Plot it against time.
3. Write a MATLAB code to compute and plot the energy spectral density of the signal. You can define the frequency vector from  $(-N/2:N/2-1)*fs/N$  for the frequency domain plot.
4. What are the frequencies at which most of the energy of the signal is concentrated?

### **Code:**

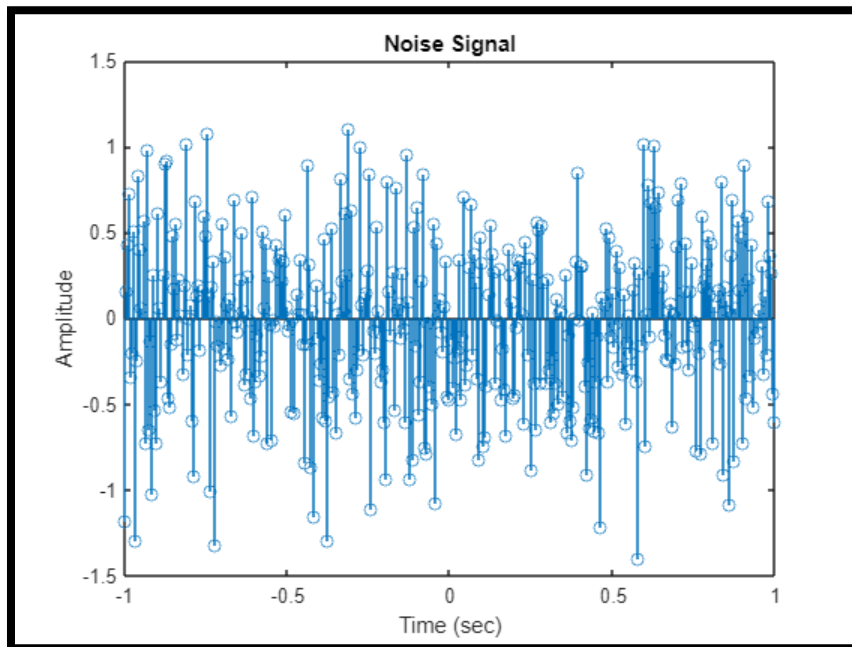
```
% Define sampling frequency and time increment
fs = 200;
dt = 1/fs;

% Define time vector
t = -fs/2:dt:fs/2-dt;

% Generate sinusoidal signal and plotting
x = 5*sin(2*pi*2*t);
stem(t, x);
xlim([-1 1]);
ylim([-6 6]);
xlabel('Time (sec)');
ylabel('Amplitude');
title('Input Signal');
```



```
% Add uniform noise
noise = 0.5*randn(size(t));
stem(t, noise);
xlim([-1 1]);
xlabel('Time (sec)');
ylabel('Amplitude');
title('Noise Signal');
```

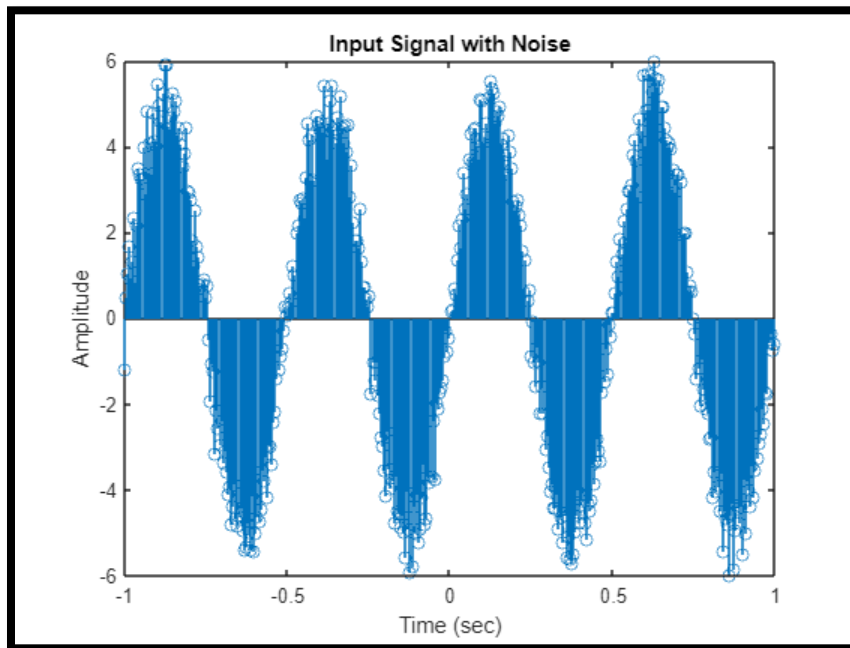


```
%subplot(5,1,3)
x_noise = x + noise;
% Plot signal against time
stem(t, x_noise);
```

```

xlim([-1 1]);
xlabel('Time (sec)');
ylabel('Amplitude');
title('Input Signal with Noise');

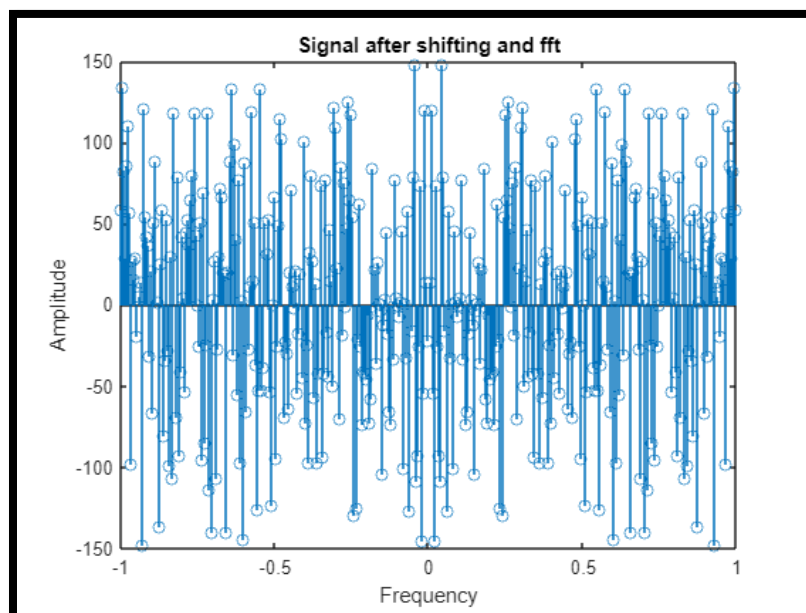
```



```

%subplot(5,1,4)
% Compute FFT and shift it
X = fftshift(fft(x_noise));
stem(t, X);
xlim([-1 1]);
xlabel('Frequency');
ylabel('Amplitude');
title('Signal after shifting and fft');

```



```

% Compute magnitude spectrum
X = abs(X);

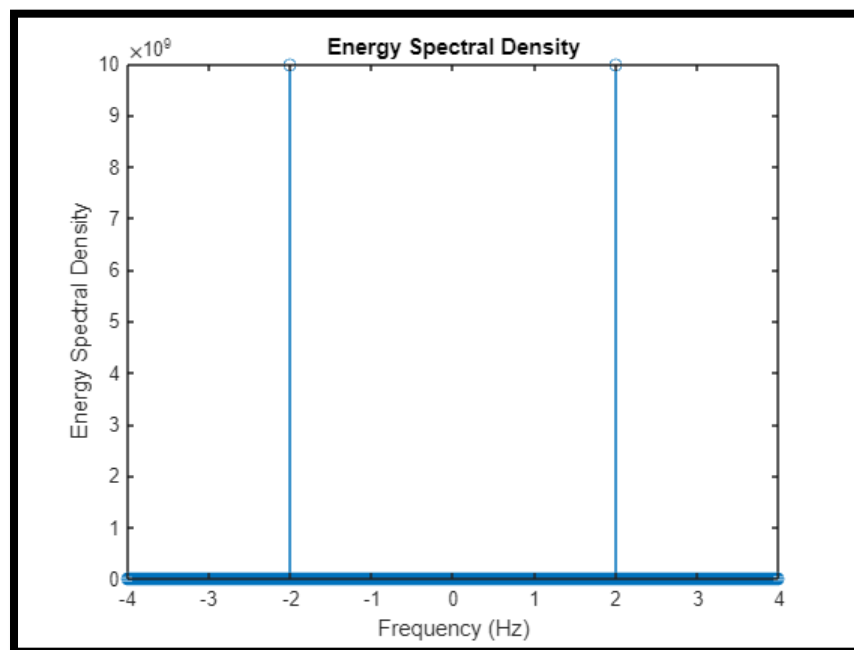
% Compute energy spectral density
X = X.^2;

% Compute total energy
total_energy = sum(X)*dt*fs/length(x);

% Compute frequency vector
f = (-length(x)/2:1:length(x)/2-1)*fs/length(x);

% Plot energy spectral density against frequency
subplot(5,1,5)
stem(f, X);
xlim([-4 4]);
xlabel('Frequency (Hz)');
ylabel('Energy Spectral Density');
title('Energy Spectral Density');

```



```

% Find frequencies at which most of the energy is concentrated
[max_energy, max_index] = max(X);
freqs_of_interest = f(max_index-5:max_index+5);

disp(['Frequencies at which most of the energy is concentrated: ',
num2str(freqs_of_interest)]);

```

Frequencies at which most of the energy is concentrated: -2.025      -2.02      -2.015      -2.01  
-2.005      -2      -1.995      -1.99      -1.985      -1.98      -1.975

***Explanation:***

*I analyzed a noisy signal in MATLAB. First, I defined the sampling frequency and the time increment. Then, I created a time vector ranging from negative half of the sampling frequency to just before the positive half. Next, I generated a sinusoidal signal and added uniform noise to it. The resulting signal was plotted against time. Afterward, I computed the Fast Fourier Transform (FFT) of the noisy signal and shifted it. The magnitude spectrum was calculated, and the energy spectral density was obtained by squaring the magnitude. The total energy of the signal was also computed. Finally, I identified the frequencies at which most of the energy was concentrated. These frequencies were determined by finding the maximum energy values in the spectrum and examining the corresponding frequency bins. The results were displayed in the console. In summary, I analyzed the input signal with noise, computed its energy spectral density, and identified significant frequency components.*

## **Exercise 2**

- Find the total energy of the signal by taking the sum of the square of its value at each time instant and multiplying the final result with the time increment.
- Find the total energy of the signal again by taking the sum of the energy spectral density and multiplying the final result with  $fs/N$ . Compare the two results.

## **Code**

```
% Define sampling frequency and time increment
fs = 200;
dt = 1/fs;

% Define time vector
t = -fs/2:dt:fs/2-dt;

% Generate sinusoidal signal
x = 5*sin(2*pi*2*t);

% Add uniform noise
x = x + 0.5*randn(size(t));

% Compute energy of the signal at each time instant
energy_at_time = sum(x.^2)*dt;

% Compute total energy of the signal
total_energy = sum(energy_at_time)*dt;
```



```

% Compute FFT and shift it
X = fftshift(fft(x));

% Compute magnitude spectrum
X = abs(X);

% Compute energy spectral density
X = X.^2;

% Compute total energy of the signal using energy spectral density
total_energy_from_spectrum = sum(X)*fs/length(x);

% Compare the two results
disp(['Total energy of the signal computed using energy spectral density: ',
num2str(total_energy_from_spectrum)]);

Total energy of the signal computed using energy spectral density: 12.739

disp(['Total energy of the signal computed using energy at each time instant: ',
num2str(total_energy)]);

Total energy of the signal computed using energy at each time instant: 12.728

```

### ***Explanation:***

*In this MATLAB code, I defined the sampling frequency and generated a sinusoidal signal with added uniform noise. I then computed the energy of the signal at each time instant and the total energy. Next, I performed the FFT and obtained the magnitude spectrum and energy spectral density. Finally, I compared the total energy computed using the energy spectral density with the total energy computed using the energy at each time instant.*

### ***Task 2 – Power Spectral Density***

The following steps can be used to find the energy spectral density of a signal 'x' using MATLAB:

1. Define  $N = \text{length}(x)$
2. Find the average power of the signal.
3. Find the Powerspectral density of signal by using `pspectrum` command.

### ***Exercise 3***

The signal  $y(t)$  is defined as below. This signal is sampled at 1000 samples per second.

$$y(t) = \begin{cases} \cos(2\pi \times 47t) + \cos(2\pi \times 219t) & ; 0 \leq t \leq 10 \\ 0 & ; \text{otherwise} \end{cases}$$

1. Plot the time domain signal  $y(t)$ .
2. Find the power spectral density of this signal.

Change the frequencies of cosines as 100 Hz and 250 Hz and explain the difference.

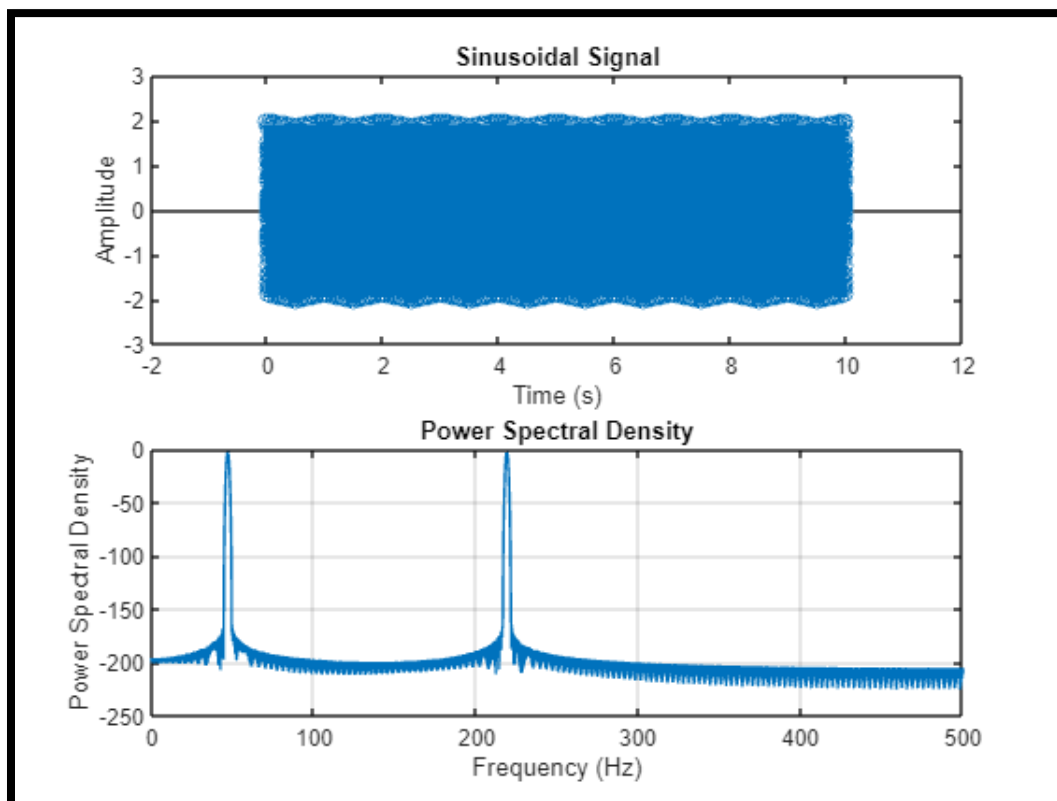
*code*

```
Fs = 1000; % Sampling frequency
t = 0:1/Fs:10;
y = cos(2*pi*47*t) + cos(2*pi*219*t);
N = length(y);
P = (norm(y)^2) / N
```

```
P = 1.0003
```

```
figure;
subplot(2,1,1);
stem(t, y);
axis([-2 12 -3 3]);
title('Sinusoidal Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2,1,2);
pspectrum(y, Fs);
title('Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power Spectral Density');
```



### **Explanation:**

*In this MATLAB code, I created a sinusoidal signal by combining two cosine waves. I computed the power of the signal and visualized it using a stem plot. Additionally, I plotted the power spectral density of the signal to show its power distribution across different frequencies.*

### **Conclusion:**

*In this Communication lab, I defined sampling frequencies, generated signals, added noise, and performed signal processing operations. I computed energy and power, analyzed signals in the time and frequency domains, and visualized the results. By utilizing various techniques such as the Fast Fourier Transform, I gained insights into the signals' characteristics. These codes allowed me to explore and understand the behavior of the signals and their energy distribution.*

**Com. Sys. Lab 5 Rubric**

**Method of Evaluation:** Executable code, Report submitted by students **Measured**

**Learning Outcomes:**

CLO1: Operate under supervision and practice hardware experiments for modulation/demodulation techniques as well as sampling of analog signals..

CLO2: Develop software simulations to observe the performance of analog and digital communications systems.

CLO4: Report desired results proofs and calculations.

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Code (CLO2)	Correct code, easily understandable with comments where necessary	Correct code but without proper indentation or comments	Slightly incorrect code with proper comments	Incorrect code with improper format and no comments	Code not submitted	
Output (CLO2)	Output correctly shown with all Figures/ Plots displayed as required and properly labelled	Most Output/ Figures/ Plots displayed with proper labels	Some Output/ Figures/ Plots displayed with proper labels OR Most Output/ Figures/ Plots displayed but without proper labels	Most of the required Output/ Figures/ Plots not displayed	Output/ Figures/ Plots not displayed	
Answers (CLO2)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Wrong Answers	
Lab Report (CLO4)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/ punctuation	Report not submitted	
Total						